ISIS 1105 Diseño y Análisis de Algoritmos
Semestre 2025-20. Proyecto – PARTE 1
Entrega: viernes, 26 de septiembre de 2025 6:00PM (por Bloque Neon)

0 OBJETIVOS

- Diseñar soluciones computacionales para problemas.
- Estimar costos de las soluciones planteadas.
- Implementar soluciones.

Se premiarán las mejores soluciones y se castigarán las peores, en cuanto a optimización y eficiencia en tiempo y espacio.

1 CONDICIONES GENERALES

El proyecto se divide en tres partes independientes entre sí. Este documento describe la PARTE I. Cada parte contiene un problema a resolver mediante soluciones implementadas en `Java` o `Python`.

Para cada problema se pide:
- Descripción de la solución.
- Análisis temporal y espacial.
- Una implementación en Java o Python

2 DESCRIPCIÓN DEL PROBLEMA

En la ciudad futurista de Tecnotown, se celebra anualmente el Festival de los Robots Creativos. Uno de los desafíos consiste en que cada robot debe distribuir una cantidad total de energía n en exactamente k celdas, de modo que la suma de las energías asignadas a cada celda corresponda al total de energía n.

El jurado del festival considera creativas únicamente aquellas celdas cuyos números contienen los dígitos 3, 6 o 9. La cantidad de puntos de creatividad obtenida por una celda depende tanto del dígito como de su posición en el número, según la siguiente tabla:

Dígito	Posición 1 (P_0)	Posición 10 (P_1)	Posición 100 (P_2)	Posición 1000 (P_3)	Posición 10000 (P_4)
3	P_0	P_1	P_2	P_3	P_4
6	$2P_0$	$2P_1$	$2P_2$	$2P_3$	$2P_4$
9	$3P_0$	$3P_1$	$3P_2$	$3P_3$	$3P_4$

Tabla 1: Puntos de creatividad asignados según el dígito y su posición

Por ejemplo, si una celda muestra el número 613, la cantidad de puntos de creatividad se calcula como sigue:

- El dígito 6 está en la posición de las centenas (P_2), por lo que aporta $2P_2$ puntos.
- El dígito 1 está en la posición de las decenas, pero no aporta puntos.
- El dígito 3 está en la posición de las unidades (P_0), por lo que aporta P_0 puntos.

Por lo tanto, la creatividad total de la celda con el número 613 es $2P_2 + P_0$.

Ejemplo:

Suponga que se debe repartir $n = 15$ unidades de energía en $k = 2$ celdas, y los valores de creatividad para cada posición son: $P_0 = 1$, $P_1 = 2$, $P_2 = 3$, $P_3 = 4$, $P_4 = 5$. El objetivo es encontrar la máxima suma posible de puntos de creatividad, considerando todas las posibles maneras de repartir la energía entre las celdas, es decir, todos los pares de números no negativos (a,b) tal que $a + b = 15$.

A continuación, se muestran algunos ejemplos de cálculo de creatividad para distintas asignaciones:

- Asignación: (6,9)
  - 6: un solo dígito (posición 0), es $6 \rightarrow 2 \times P_0 = 2 \times 1 = 2$. – 9: un solo dígito (posición 0), es $9 \rightarrow 3 \times P_0 = 3 \times 1 = 3$.
  - **Creatividad total**: 2 + 3 = 5.
- Asignación: (12,3)
  - 12: unidades: 2 (no suma), decenas: 1 (no suma), → 0 puntos.
  - 3: unidades: $3 \rightarrow P_0 = 1$.
  - **Creatividad total**: 0 + 1 = 1.
- Asignación: (3,12)
  - 3: unidades: $3 \rightarrow P_0 = 1$.
  - 12: unidades: 2 (no suma), decenas: 1 (no suma), → 0 puntos.
  - **Creatividad total**: 1 + 0 = 1.
- Asignación: (15,0)
  - 15: unidades: 5 (no suma), decenas: 1 (no suma), → 0 puntos.
  - 0: no suma.
  - **Creatividad total**: 0.

La mayor creatividad posible en este caso se obtiene con las asignaciones (6,9) o (9,6), logrando un total de 5 puntos de creatividad.

3 ENTRADA Y SALIDA DE DATOS

En todas las soluciones que se presenten, la lectura de los datos de entrada se hace por la entrada estándar; así mismo, la escritura de los resultados se hace por la salida estándar.

Puede suponer que ninguna línea de entrada tiene espacios al principio o al final, y que los datos que se listan en cada línea están separados por exactamente un espacio.

A continuación, se establecen parámetros que definen su tamaño y formato de lectura de los datos, tanto de entrada como de salida.

Descripción de la entrada

La primera línea de entrada especifica el número de casos de prueba que contiene el archivo. El programa debe terminar su ejecución, una vez termine de resolver la cantidad de casos de prueba dados por este número.

Cada una de las siguientes líneas contienen un caso de prueba descrito por siete números enteros. El primer número entero k ($1 \leq k \leq 10^4$) representa el número de celdas en las que se debe repartir la energía. El segundo número entero *n* indica la suma total de energía que se debe repartir ($1 \leq n \leq 10^5$). Los siguientes cinco números enteros P_0, P_1, P_2, P_3, P_4, indican la creatividad asignada a cada posición de dígito ($1 \leq P_i \leq 10^5$).

Descripción de la salida

Para cada caso de prueba, la salida deben ser una sola línea con el máximo de creatividad posible.

Ejemplo de entrada / salida

Entrada	Salida
3 2 15 1 2 3 4 5 3 57 1 2 3 4 5 2 600 3 1 2 4 3	5 11 14

Nota: Se van a diseñar casos de prueba para valores de n, k mucho más grandes y dentro de los valores establecidos en el enunciado. Los casos mostrados en este documento son demostrativos de la estructura de entrada/salida esperada.

5 ENTREGABLES

El proyecto puede desarrollarse por grupos de hasta dos estudiantes de la misma sección. La entrega se hace por bloque neon (una sola entrega por grupo de trabajo).

El grupo debe entregar, por bloque neon, un archivo de nombre `proyectoDalgoP1.zip`. Este archivo es una carpeta de nombre `proyectoDalgoP1`, comprimida en formato `.zip`, dentro de la cual hay archivos fuente de soluciones propuestas y archivos que documentan cada una de las soluciones.

5.1 Archivos fuente de soluciones propuestas

Todos los programas implementados en *Java* o en Python

Para el problema:

- Entregar un archivo de código fuente en *Java* (`.java`) o python (`.py`) con su código fuente de la solución que se presenta.
- Incluir como encabezado de cada archivo fuente un comentario que identifique el (los) autor(es) de la solución.
- Denominar `ProblemaP1.java` o `ProblemaP1.py` el archivo de la solución que se presente.

Nótese que, si bien puede utilizarse un *IDE* como *Eclipse* o Spyder durante el desarrollo del proyecto, la entrega requiere incluir solo un archivo por cada solución. El archivo debe poderse compilar y ejecutar independientemente (sin depender de ninguna estructura de directorios, librerías no estándar, etc.).

5.2 Archivos que documentan la solución propuesta

La solución al problema debe acompañarse de un archivo de máximo 3 páginas que la documente, con extensión `.pdf`. El nombre del archivo debe ser el mismo del código correspondiente (`ProblemaP1.pdf`).

Un archivo de documentación debe contener los siguientes elementos:

0 Identificación
Nombre de autor(es)

1 Algoritmo de solución
Explicación del algoritmo elegido. Si hubo alternativas de implantación diferentes, explicar por qué se escogió la que se implementó. Generar al menos una gráfica que apoye la explicación del algoritmo implementado. No se debe copiar y pegar código fuente como parte de la explicación del algoritmo.

2 Análisis de complejidades espacial y temporal
Cálculo de complejidades y explicación de estas.

La nota del informe corresponde a un 50% de la nota total de la entrega del proyecto, solamente si se entrega el código fuente de la solución implementada. En caso de no entregar el código fuente, no se hará evaluación del informe y la nota del proyecto será cero.

Además de la pertinencia del texto como explicación de la solución implementada, se evaluará la calidad en la redacción del texto y en el diseño de las gráficas. Se evaluará también que la explicación del algoritmo integre los conceptos relacionados con las técnicas de diseño de algoritmos cubiertas en el curso.

Téngase en cuenta que los análisis de 2 tienen sentido en la medida que la explicación de 1 sea clara y correcta. No se está exigiendo formalismo a ultranza, pero sí que, como aplicación de lo estudiado en el curso, se pueda describir un algoritmo de manera correcta y comprensible.

5.3 Consideraciones sobre la entrega

Lea bien las recomendaciones de entrada/salida. Su proyecto será evaluado, en gran parte, por una máquina: si en la ejecución del programa el formato de salida no coincide perfectamente con lo requerido, la calificación de la ejecución será cero, sin importar la cantidad de código que haya desarrollado.

La forma en que se presenta la documentación debe respetarse, aunque su evaluación no sea tan automática como la del software. Hay que nombrar los archivos como se espera que se nombren, comprimirlos como se pide que se compriman, etc. Cualquier desviación en cuanto a lo que se pide produce deméritos en la calificación final.

El software se evalúa desde la línea de comandos de Linux. No hay problema en desarrollar en cualquier otro sistema (Mac, Unix, ...) siempre que se produzca código estándar en java o python. Cada problema se debe resolver en un (1) archivo.